

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-332786

(43)Date of publication of application : 30.11.2000

(51)Int.Cl.

H04L 12/28

H04L 12/56

H04M 3/00

(21)Application number : 11-140908

(71)Applicant : NIPPON TELEGR & TELEPH  
CORP <NTT>

(22)Date of filing : 21.05.1999

(72)Inventor : UGA MASANORI  
SHIOMOTO KOHEI

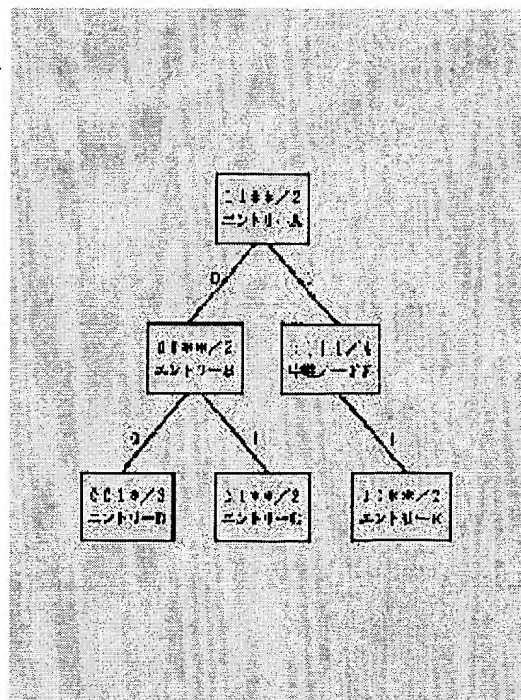
## (54) METHOD FOR RETRIEVAL FROM ROUTING TABLE

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To decrease the number of bits needed for one entry of the routing table by storing information on the difference between the IP address of a master node and the IP address of a slave node in a Patricia tree in the routing table and retrieving routing information on the destination IP address that an IP packet has.

**SOLUTION:** A deformed tree is constituted, by converting an 8-bit length Patricia tree into 4-bit length difference information. In the routing table corresponding to the deformed tree, the difference 01\*\* between an entry B and an entry A as its parent node is stored as difference information of a table of, for example, the entry B and the number 2 of effective bits of the

difference is stored. Furthermore, the difference 11111111 between the entries A and E cannot be represented with four bits, so that a repeating node F is inserted between the entries A and E. Consequently, the deformed tree obtained by converting the Patricia tree and the routing table can be generated to reduce the memory quantity.



## LEGAL STATUS

[Date of request for examination] 17.10.2001

[Date of sending the examiner's decision of rejection] 16.12.2003

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

## \* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to an economical search method especially about the search method of the routing table at the time of the node of IP communication network performing routing.

[0002]

[Description of the Prior Art] The die length of PURIFIKUSU of the IP address of the IP address itself and its node is set to the node of IP communication network, respectively. Drawing 1 is drawing showing the conventional example, and drawing 1 (a) shows the example of the conventional routing table corresponding to [ in drawing 1 (b) ] the PATORISHIA tree of drawing 1 (a) for the example of a PATORISHIA tree. In this example, Entry X (X=A, B, ...) is an identifier which distinguishes a node uniquely, it is the bit length  $N=8$  of an IP address, and those values are [ \* expresses the part which does not belong to PURIFIKUSU among IP addresses and ] 0 or 1.

[0003] The routing table search method using a PATORISHIA tree is as follows. The bit (here 2) equivalent to the die length of high order PURIFIKUSU of the IP address and destination address which a certain node (for example, node of the entry A of drawing) shows is measured. Retrieval is ended as information which searches for information, such as the NeXT hop which the table corresponding to the node which was in agreement immediately before has when not in agreement, and when in agreement, information, such as the NeXT hop which the table corresponding to the node (node of Entry A) has, is read and recorded.

[0004] Next, when it moves to the child node (drawing Entry B or an entry (E)), and retrieval is continued, when a child node exists, and a child node does not exist with the value with reference to the bit (here 3rd bit) of the 1st [ + ] die length of high order PURIFIKUSU of a destination IP address, retrieval is ended and it considers as the information which searches for information, such as the NeXT hop which recorded.

[0005] The conventional routing table stored the information which shows the die length of PURIFIKUSU of the IP address of the node on a PATORISHIA tree itself, and the IP address of the node, and the information on other (for example, the index to degree entry, NeXT hop, etc.) about one entry, as shown in drawing 1 (b). In such conventional routing table, R bits showing information, such as N bit (at IPv4, it is 128 in 32 bits and IPv6 bit) which expresses an IP address about one entry, a  $\log_2 N$  bit showing the die length of PURIFIKUSU, and the NeXT hop, were needed. At IPv6, it is 128. Since it was a bit, when the conventional routing table was created and this was stored in memory, there was a problem of  $135+R$  bits per one entry having been required, and requiring the amount of memory great as the whole.

[0006]

[Problem(s) to be Solved by the Invention] In view of the above troubles, the purpose of this invention reduces the number of bits required for per one entry of routing table, and is to offer an economical routing table search method.

[0007]

[Means for Solving the Problem] in order that the routing table search method of this invention may attain the above-mentioned purpose -- routing table -- the difference of the IP address of the parent node on a PATORISHIA tree, and the IP address of the child node of that -- information -- memorizing -- this difference -- it is characterized by retrieving the routing information on the destination IP address which an IP packet has based on information.

[0008] the routing table search method of this invention -- setting -- said difference -- when information is from  $j$  bits of high orders of the destination IP address which an IP packet has to a high order  $j+k$  bit, it is desirable to memorize  $j$  and  $k$ . moreover, said difference -- the case where information is the bit length below  $M$  bit shorter than the bit length of an IP address -- difference -- information -- the routing table of a child node -- memorizing -- said difference -- the case where information exceeds  $M$  bit -- between a parent node and child nodes -- every  $M$  bit -- a junction node -- preparing -- difference -- it is desirable to memorize information to the routing table of each junction node and the routing table of a child node.

[0009] the difference of the IP address which the parent node which replaces routing table with an IP address according to the routing table search method of such this invention, and is on a PATORISHIA tree has, and the IP address which the child node of the parent node has -- only information is memorized. this difference -- in expressing as information  $j$  bits of from high orders of the destination IP address of  $N$  bit length which an IP packet has to a high order  $j+k$  bit, a  $\log_2 N$  bit is provided about  $j$ , a  $\log_2 M$  bit tooth space is provided for a table about  $k$ , and it memorizes  $j$  and  $k$  on it. In this case,  $j$  may be stored in a register.

[0010] this difference -- when information is larger than the tooth space provided for routing table, a junction node is added between the parent node of routing table, and a child node. this junction node -- difference -- it has only  $M$  bits of informational high orders. difference -- information --  $M$  bit and 2 -- the case where it cannot express with  $M$  bits .. a pieces and .. a junction node -- two pieces, three ..  $a+1$  piece, and .. as -- it adds continuously. namely, difference -- the case where information cannot express with  $M$  bit -- the difference between a parent node and a child node --  $M$  bits of informational high orders -- the 1st junction node -- storing -- a degree -- the difference between the 1st junction node and a child node --  $M$  bits of informational high orders are stored in the 2nd junction node. adding a junction node like the following -- the difference between a parent node and a child node -- information is stored in two or more junction nodes and child nodes.

[0011] The approach of searching the destination IP address which an IP packet has using routing table Longest-first PURIFIKUSU is searched using the tree which transformed the PATORISHIA tree. the difference of the child node and parent node which routing table has in case it moves from the parent node of a PATORISHIA tree to a child node -- information -- and the difference -- the register showing information being from  $j$  bits of high orders of an IP address to a high order  $j+k$  bit, and routing table information -- from  $j$  bits of high orders of read-out and a destination IP address up to a high order  $j+k$  bit -- difference -- it collates with information.

[0012] the information which a table should have about one entry by this -- difference --  $M$  bit and difference showing information -- it is  $\log_2 N + \log_2 M$  bit and  $R$  bits of other information for information to express even what bit from the what bit high order of an IP address, and the width of face of routing table can be reduced remarkably.

[0013]

[Embodiment of the Invention] Next, the example of this invention is explained. Drawing 2 is drawing showing the deformation tree by the search method of this invention, and drawing 3 is drawing showing the example of the routing table corresponding to the deformation tree of drawing 2. the PATORISHIA tree of the bit length  $N=8$  which showed drawing 2 in detail to drawing 1 (a) -- this invention -- the difference of 4 bit length -- the deformation tree changed into information is shown.

[0014] a child node -- the difference of the child node and parent node -- since information is stored -- the difference of the table of for example, the entry B -- memorizing to information the difference of Entry B and the entry A which is the parent node, i.e.,  $01^{**}$ , the number of effective bits of difference

memorizes 2. moreover, the difference which the difference of Entry A and Entry E is 111111, and was prepared in the routing table of this example -- 4 bits which is a tooth space for memorizing information -- this difference -- since information cannot be expressed, the junction node F is inserted between Entry A and Entry E. the junction node F -- the difference of Entry A and Entry E -- the difference prepared in routing table among information -- 1111 for 4 bits which are a tooth space for memorizing information - difference -- it memorizes as information and the four effective bits are memorized. furthermore, the difference of the junction node F and Entry E -- 11\*\* -- it is -- the difference -- the difference by which information was prepared in routing table -- since it can express by 4 bits which is a tooth space for memorizing information -- the difference of the entry E of routing table -- 11\*\* is memorized to information and two effective bits are memorized. By performing the same processing as the following, the deformation tree and routing table which were changed from the PATORISHIA tree can be created.

[0015] Each item of the other information on the routing table of drawing 3 has following semantics. 0E is a value which shows the existence of the existence of the child node whose retrieval bit is 0, for example, when it exists, it has 1, and when it does not exist, it has the value of 0. The zero address shows the address with which the entry of the child node exists, when the child node whose retrieval bit is 0 exists. The number of bits which this term needs is decided with the number of the entries of a table. 1E is a value which shows the existence of the existence of the child node whose retrieval bit is 1, for example, when it exists, it has 1, and when it does not exist, it has the value of 0. The single address shows the address with which the entry of the child node exists, when the child node whose retrieval bit is 1 exists. The number of bits which this term needs is decided with the number of the entries of a table. Here, a retrieval bit means the next bit of the bit of the last referred to by the node among the destination IP addresses of the IP packet which reached the node, and, thereby, branching to a child node is decided.

[0016] Drawing 4 is drawing showing the example of the routing table search method of this invention using the example of routing table shown in drawing 2 and 3. j -- the j-th bit of the high order of a destination IP address -- expressing -- the difference of a table -- the bit position of the beginning of the destination IP address in comparison with information is shown. First, as initialization, it is referred to as j= 1, the address of the root entry of a tree is substituted for A which shows the address of routing table, and the fixed root is substituted for NH which shows the NeXT hop etc. (step 1).

[0017] next, the contents of the address A of a table -- read in and difference -- information, the number of effective bits, the value of 0E, the value of the zero address, the value of 1E, and the value of the single address -- Variables sabun and yuukou and 0 -- en, 1en, and 0ad And 1ad It substitutes (step 2). Next, it is [ the j-th bit of the high order of a destination IP address to eye a high order j+yuukou bit, and ] sabun. It compares (step 3). When not in agreement as a result of this comparison, retrieval is ended, and it considers as the information which acquired the information stored in NH by this retrieval (step 4). It is 0en which substitutes the NeXT hop information on Address A for NH when in agreement as a result of the comparison of step 3, substitutes j+yuukou +1 for j (step 5), reads the j-th bit of the high order of a destination IP address, investigates 0 or 1 (step 6), and corresponds to the value. Or 1en Existence of a child node is checked (step 7 or 8). It is A=0ad when a child node exists. Or A=1ad It carries out and goes to a child node (step 9 or 10). It returns to step 2 below and searches similarly. When a child node does not exist by step 7 or 8, retrieval is ended, and it considers as the information which acquired the information stored in NH by this retrieval (step 4).

[0018] the time of memorizing j in routing table and moving to other nodes, although how not to memorize j to routing table in the above-mentioned example was shown -- a table to j -- eye the high order j+k bit from the j-th bit of the high order of read in and a destination IP address, and difference -- you may make it compare information

[0019] moreover -- since the j-th bit of a destination IP address is investigated and it moves from it, when moving from a node to a child node -- not necessarily -- difference -- it is not necessary to have the j-th bit of the high order as information therefore, difference -- you may make it have the j+1st bits of the high order of the IP address which a child node shows as information as first bit for example, it is shown in drawing 5 -- as -- Node U -- difference -- it has 01\*\*/2 as information, and difference of Node

U and a parent node is set to  $001^{*}/3$  which added the reference bit 0.

[0020]

[Effect of the Invention] as mentioned above -- according to the routing table search method of this invention -- the number of bits required for one entry of routing table -- the conventional approach -- bit length N of an IP address, and the difference with an IP address shorter than bit length N although expressed  $N + \log_2 N + R$  using informational bit length R in addition to this by the approach of this invention to this -- it becomes  $N + \log_2 N + R$  using informational bit length M. For example, at IPv6, although 182 bits of information, such as 32,000 and the NeXT hop, of the number of entries of routing table were required for the routing table which is 15 bits per one entry and by the conventional approach, it can be managed with 58 bits by the approach of this invention. When using memory with 16-bit width of face, to having been required for 12 pieces per one entry and by the conventional approach, by the approach of this invention, it will end with four pieces and the amount of memory can be reduced to one third by this invention.

---

[Translation done.]

PAT-NO: JP02000332786A

DOCUMENT-IDENTIFIER: JP 2000332786 A

TITLE: METHOD FOR RETRIEVAL FROM ROUTING  
TABLE

PUBN-DATE: November 30, 2000

INVENTOR-INFORMATION:

|                 |         |
|-----------------|---------|
| NAME            | COUNTRY |
| UGA, MASANORI   | N/A     |
| SHIOMOTO, KOHEI | N/A     |

ASSIGNEE-INFORMATION:

|                                   |         |
|-----------------------------------|---------|
| NAME                              | COUNTRY |
| NIPPON TELEGR & TELEPH CORP <NTT> | N/A     |

APPL-NO: JP11140908

APPL-DATE: May 21, 1999

INT-CL (IPC): H04L012/28, H04L012/56 , H04M003/00

ABSTRACT:

PROBLEM TO BE SOLVED: To decrease the number of bits needed for one entry of the routing table by storing information on the difference between the IP address of a master node and the IP address of a slave node in a Patricia tree in the routing table and retrieving routing information on the destination IP address that an IP packet has.

SOLUTION: A deformed tree is constituted, by converting an 8-bit length Patricia tree into 4-bit length difference information. In the routing table corresponding to the deformed tree, the difference 01\*\*

between an entry B and  
an entry A as its parent node is stored as difference  
information of a table  
of, for example, the entry B and the number 2 of effective  
bits of the  
difference is stored. Furthermore, the difference 11111111  
between the entries  
A and E cannot be represented with four bits, so that a  
repeating node F is  
inserted between the entries A and E. Consequently, the  
deformed tree obtained  
by converting the Patricia tree and the routing table can  
be generated to  
reduce the memory quantity.

COPYRIGHT: (C)2000,JPO



(19)日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-332786

(P2000-332786A)

(43)公開日 平成12年11月30日 (2000. 11. 30)

| (51)Int.Cl. <sup>7</sup> | 識別記号 | F I           | テーマコード* (参考) |           |
|--------------------------|------|---------------|--------------|-----------|
| H 0 4 L 12/28            |      | H 0 4 L 11/00 | 3 1 0 D      | 5 K 0 3 0 |
| 12/56                    |      | H 0 4 M 3/00  | D            | 5 K 0 3 3 |
| H 0 4 M 3/00             |      | H 0 4 L 11/20 | 1 0 2 D      | 5 K 0 5 1 |

審査請求 未請求 請求項の数 3 O L (全 5 頁)

(21)出願番号 特願平11-140908

(22)出願日 平成11年5月21日 (1999. 5. 21)

(71)出願人 000004226

日本電信電話株式会社

東京都千代田区大手町二丁目3番1号

(72)発明者 宇賀 雅則

東京都新宿区西新宿3丁目19番2号 日本  
電信電話株式会社内

(72)発明者 塩本 公平

東京都新宿区西新宿3丁目19番2号 日本  
電信電話株式会社内

(74)代理人 100059258

弁理士 杉村 暁秀 (外 1 名)

Fターム(参考) 5K030 GA05 HA08 HC14 KA05 LB05

5K033 AA04 DA16 DB12 EC04

5K051 AA05 CC02 EE01 FF16 KK05

## (54)【発明の名称】 ルーティングテーブル検索方法

## (57)【要約】

【課題】 IP通信網のノードがルーティングを行うためのルーティングテーブルの1エン트리あたりに必要なビット数を減らし、経済的なルーティングテーブル検索方法を提供する。

【解決手段】 ルーティングテーブルに、パトリシアツリー上の親ノードのIPアドレスとその子ノードのIPアドレスとの差分情報を記憶し、この差分情報に基づいて宛先IPアドレスのルーティング情報を検索する。差分情報が、宛先IPアドレスの上位jビットから上位j+kビットまでである場合に、j及びkを記憶する。また、差分情報がIPアドレスのビット長より短いMビット以下のビット長である場合は差分情報を子ノードに記憶し、差分情報がMビットを超える場合は親ノードと子ノードとの間にMビット毎に中継ノードを設け、差分情報をそれぞれの中継ノード及び子ノードに記憶する。

| エン트리 | 差分情報<br>(4ビット) | 有効ビット数<br>(2ビット) | 0E | 0アドレス   | 1E | 1アドレス   | ネクスト<br>ホップ等 |
|------|----------------|------------------|----|---------|----|---------|--------------|
| A    | 11**           | 2                | 1  | アドレス(B) | 1  | アドレス(F) |              |
| B    | 01**           | 2                | 1  | アドレス(D) | 1  | アドレス(C) |              |
| F    | 1111           | 4                | 0  |         | 1  | アドレス(E) |              |
| C    | 11**           | 2                | 0  |         | 0  |         |              |
| D    | 001*           | 3                | 0  |         | 0  |         |              |
| E    | 11**           | 2                | 0  |         | 0  |         |              |

## 【特許請求の範囲】

【請求項1】 IPパケットが持つ宛先IPアドレスからルーティング情報を検索するために用いるルーティングテーブルを検索する方法において、ルーティングテーブルに、パトリシアツリー上の親ノードのIPアドレスとその子ノードのIPアドレスとの差分情報を記憶し、この差分情報に基づいてIPパケットが持つ宛先IPアドレスのルーティング情報を検索することを特徴とするルーティングテーブル検索方法。

【請求項2】 前記差分情報が、IPパケットが持つ宛先IPアドレスの上位jビットから上位j+kビットまでである場合に、j及びkを記憶することを特徴とする請求項1に記載のルーティングテーブル検索方法。

【請求項3】 前記差分情報がIPアドレスのビット長より短いMビット以下のビット長である場合は差分情報を子ノードのルーティングテーブルに記憶し、前記差分情報がMビットを超える場合は親ノードと子ノードとの間にMビット毎に中継ノードを設け、差分情報をそれぞれの中継ノードのルーティングテーブル及び子ノードのルーティングテーブルに記憶することを特徴とする請求項1に記載のルーティングテーブル検索方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明はIP通信網のノードがルーティングを行う際のルーティングテーブルの検索方法に関するものであり、特に経済的な検索方法に関する。

## 【0002】

【従来の技術】IP通信網のノードには、それぞれIPアドレスそのもの及びそのノードのIPアドレスのプリフィクスの長さが定められている。図1は従来の例を示す図であり、図1(a)はパトリシアツリーの例を、図1(b)は図1(a)のパトリシアツリーに対応する従来のルーティングテーブルの例を示す。この例では、IPアドレスのビット長N=8であり、エントリーX(X=A, B,...)はノードを一意的に区別する名前であり、\*はIPアドレスのうちプリフィクスに属さない部分を表すもので、それらの値は0又は1である。

【0003】パトリシアツリーを用いるルーティングテーブル検索方法は以下のとおりである。或るノード(例えば図のエントリーAのノード)が示すIPアドレスと宛先アドレスとの上位プリフィクスの長さに相当するビット(ここでは2)を比較する。一致しない場合は、直前に一致したノードに対応するテーブルが持つネクストホップ等の情報を求める情報として検索を終了し、一致した場合は、そのノード(エントリーAのノード)に対応するテーブルが持つネクストホップ等の情報を讀込んで記録する。

【0004】次に、宛先IPアドレスの上位プリフィクスの長さ+1番目のビット(ここでは3番目のビット)

を参照し、その値によって、子ノードが存在する場合はその子ノード(図ではエントリーB又はエントリーE)へ移って検索を続け、子ノードが存在しない場合は検索を終了し、記録したネクストホップ等の情報を求める情報とする。

【0005】従来のルーティングテーブルは、図1(b)に示すように、1エントリーについて、パトリシアツリー上のノードのIPアドレスそのもの、そのノードのIPアドレスのプリフィクスの長さを示す情報及びその他の情報(例えば、次エントリーへのインデックス、ネクストホップ等)を格納していた。このような従来のルーティングテーブルにおいては、1エントリーについて、IPアドレスを表すNビット(例えばIPv4では32ビット、IPv6では128ビット)、プリフィクスの長さを表す $\log_2 N$ ビット及びネクストホップ等の情報を表すRビットを必要としていた。IPv6では128ビットであるため、従来のルーティングテーブルを作成すると、これをメモリーに格納する場合、1エントリー当たり $135+R$ ビットが必要であり、全体としては多大なメモリー量を要するという問題があった。

## 【0006】

【発明が解決しようとする課題】本発明の目的は、上述のような問題点に鑑み、ルーティングテーブルの1エントリー当たりに必要なビット数を減らし、経済的なルーティングテーブル検索方法を提供することにある。

## 【0007】

【課題を解決するための手段】本発明のルーティングテーブル検索方法は、上記の目的を達成するため、ルーティングテーブルに、パトリシアツリー上の親ノードのIPアドレスとその子ノードのIPアドレスとの差分情報を記憶し、この差分情報に基づいてIPパケットが持つ宛先IPアドレスのルーティング情報を検索することを特徴とする。

【0008】本発明のルーティングテーブル検索方法においては、前記差分情報が、IPパケットが持つ宛先IPアドレスの上位jビットから上位j+kビットまでである場合に、j及びkを記憶することが望ましい。また、前記差分情報がIPアドレスのビット長より短いMビット以下のビット長である場合は差分情報を子ノードのルーティングテーブルに記憶し、前記差分情報がMビットを超える場合は親ノードと子ノードとの間にMビット毎に中継ノードを設け、差分情報をそれぞれの中継ノードのルーティングテーブル及び子ノードのルーティングテーブルに記憶することが望ましい。

【0009】このような本発明のルーティングテーブル検索方法によれば、ルーティングテーブルは、IPアドレスに代えて、パトリシアツリー上にある親ノードが持つIPアドレスとその親ノードの子ノードが持つIPアドレスとの差分情報のみを記憶する。この差分情報として、IPパケットが持つNビット長の宛先IPアドレス

の上位jビットから上位j+kビットまでを表す場合には、テーブルに、jについて $\log_2 N$ ビット、kについて $\log_2 M$ ビットのスペースを設けて、j及びkを記憶する。この場合、jはレジスタに格納してもよい。

【0010】この差分情報がルーティングテーブルに設けられたスペースより大きい場合は、ルーティングテーブルの親ノードと子ノードとの間に中継ノードを追加する。この中継ノードは、差分情報の上位Mビットのみを持つ。差分情報がMビット、2Mビット、...a個、...で表すことができない場合は、中継ノードを2個、3個、...a+1個、...というように、連続して追加する。即ち、差分情報がMビットで表すことができない場合は、親ノードと子ノードとの間の差分情報の上位Mビットを第1の中継ノードに格納し、次に第1の中継ノードと子ノードとの間の差分情報の上位Mビットを第2の中継ノードに格納する。以下同様に中継ノードを追加することにより、親ノードと子ノードとの間の差分情報を複数の中継ノード及び子ノードに格納する。

【0011】ルーティングテーブルを用いてIPパケットが持つ宛先IPアドレスを検索する方法は、パトリシアツリーを変形したツリーを利用して最長一致プリフィクスを検索し、パトリシアツリーの親ノードから子ノードへ移る際に、ルーティングテーブルが持つ子ノードと親ノードとの差分情報、及びその差分情報がIPアドレスの上位jビットから上位j+kビットまでであることを表すレジスタ及びルーティングテーブル情報を読み出し、宛先IPアドレスの上位jビットから上位j+kビットまでを差分情報と照合する。

【0012】これにより、1エントリーについてテーブルが持つべき情報は、差分情報を表すMビット、差分情報がIPアドレスの上位何ビット目から何ビット目までかを表すための $\log_2 N + \log_2 M$ ビット、及び他の情報Rビットだけであり、ルーティングテーブルの幅を著しく削減することができる。

【0013】

【発明の実施の形態】次に本発明の実施例を説明する。図2は本発明の検索方法による変形ツリーを示す図であり、図3は図2の変形ツリーに対応するルーティングテーブルの例を示す図である。図2は、詳しくは、図1(a)に示したビット長N=8のパトリシアツリーを、本発明により4ビット長の差分情報に変換した変形ツリーを示している。

【0014】子ノードにはその子ノードと親ノードとの差分情報を格納するので、例えばエントリーBのテーブルの差分情報にはエントリーBとその親ノードであるエントリーAとの差分、即ち01\*\*を記憶し、差分の有効ビット数は2を記憶する。また、エントリーAとエントリーEとの差分は111111であり、この例のルーティングテーブルに設けられた差分情報を記憶するためのスペースである4ビットではこの差分情報を表せない

ため、エントリーAとエントリーEとの間に中継ノードFを挿入する。中継ノードFはエントリーAとエントリーEとの差分情報のうちルーティングテーブルに設けられた差分情報を記憶するためのスペースである4ビット分の1111のみを差分情報として記憶し、その有効ビット数4を記憶する。更に、中継ノードFとエントリーEとの差分は11\*\*であり、その差分情報は、ルーティングテーブルに設けられた差分情報を記憶するためのスペースである4ビットで表現できるので、ルーティングテーブルのエントリーEの差分情報には11\*\*を記憶し、有効ビット数2を記憶する。以下同様の処理を行うことにより、パトリシアツリーから変換した変形ツリー及びルーティングテーブルを作成することができる。

【0015】図3のルーティングテーブルの他情報の各項は以下の意味を有する。0Eは、検索ビットが0である子ノードの存在の有無を示す値であり、例えば、存在する場合は1、存在しない場合は0の値を持つ。0アドレスは、検索ビットが0である子ノードが存在する場合にその子ノードのエントリーが存在するアドレスを示す。この項が必要とするビット数はテーブルのエントリーの数によって決まる。1Eは、検索ビットが1である子ノードの存在の有無を示す値であり、例えば、存在する場合は1、存在しない場合は0の値を持つ。1アドレスは、検索ビットが1である子ノードが存在する場合にその子ノードのエントリーが存在するアドレスを示す。この項が必要とするビット数はテーブルのエントリーの数によって決まる。ここで、検索ビットとは、そのノードに到着したIPパケットの宛先IPアドレスのうち、そのノードで参照した最後のビットの次のビットを意味し、これにより子ノードへの分岐が決まる。

【0016】図4は図2及び3に示したルーティングテーブルの例を用いる本発明のルーティングテーブル検索方法の実施例を示す図である。jは宛先IPアドレスの上位jビット目を表し、テーブルの差分情報と比較する宛先IPアドレスの最初のビット位置を示す。先ず初期化として、j=1とし、ルーティングテーブルのアドレスを示すAにツリーのルートエントリーのアドレスを代入し、ネクストホップ等を示すNHに既定のルートを代入する(ステップ1)。

【0017】次に、テーブルのアドレスAの中身を読み込み、差分情報、有効ビット数、0Eの値、0アドレスの値、1Eの値及び1アドレスの値を、変数sabun、yuukou、0en、1en、0ad及び1adに代入する(ステップ2)。次に、宛先IPアドレスの上位jビット目から上位j+yuukouビット目までとsabunとを比較する(ステップ3)。この比較の結果一致しない場合は検索を終了し、NHに格納されている情報をこの検索によって取得した情報とする(ステップ4)。ステップ3の比較の結果一致する場合はアドレスAのネクストホップ情報をNHに代入し、jにj+yuukou+1を代入し(ステップ

5

5)、宛先IPアドレスの上位jビット目を読出して0か1かを調べ(ステップ6)、その値に該当する0en又は1enによって子ノードの存在を確認する(ステップ7又は8)。子ノードが存在する場合はA=0ad又はA=1adとして子ノードへ行く(ステップ9又は10)。以下ステップ2へ戻って同様に検索を行う。ステップ7又は8で子ノードが存在しない場合は検索を終了し、NHに格納されている情報をこの検索によって取得した情報とする(ステップ4)。

【0018】上記の実施例においてはjをルーティングテーブルに記憶しない方法を示したが、jをルーティングテーブル中に記憶し、他のノードに移った時にテーブルからjを読み込み、宛先IPアドレスの上位jビット目から上位j+kビット目と差分情報とを比較するようにしてもよい。

【0019】また、ノードから子ノードへ移る場合に宛先IPアドレスのjビット目を調べて移るので、必ずしも差分情報として上位jビット目を持つ必要はない。そのため、差分情報として子ノードが示すIPアドレスの上位j+1ビット目を最初のビットとして持つようにしてもよい。例えば、図5に示すように、ノードUは差分情報として01\*\*\*/2を持ち、ノードUと親ノードとの差分は参照ビット0を加えた001\*\*\*/3とする。

【0020】

【発明の効果】上述のように、本発明のルーティングテーブル検索方法によれば、ルーティングテーブルの1エントリーに必要なビット数は、従来の方法では、IPアドレスのビット長N及びその他情報のビット長Rを用いて、 $N + \log_2 N + R$ と表されるが、これに対して本発明

6

の方法では、IPアドレスのビット長Nより短い差分情報のビット長Mを用いて、 $N + \log_2 N + R$ となる。例えば、IPv6ではルーティングテーブルのエントリー数が32,000、ネクストホップ等の情報が15ビットであるルーティングテーブルは、1エントリー当たり、従来の方法では182ビット必要であったが、本発明の方法では58ビットで済む。16ビット幅を持つメモリーを用いる場合、1エントリー当たり、従来の方法では12個必要であったのに対して、本発明の方法では4個で済むことになり、本発明によりメモリー量を1/3に減らすことができる。

【図面の簡単な説明】

【図1】従来のパトリシアツリー及びルーティングテーブルの例を示す図である。

【図2】本発明の検索方法による変形ツリーの例を示す図である。

【図3】図2の変形ツリーに対応するルーティングテーブルの例を示す図である。

【図4】図2及び3に示したルーティングテーブルの例を用いる本発明のルーティングテーブル検索方法の実施例を示す図である。

【図5】本発明の検索方法による変形ツリーの他の例を示す図である。

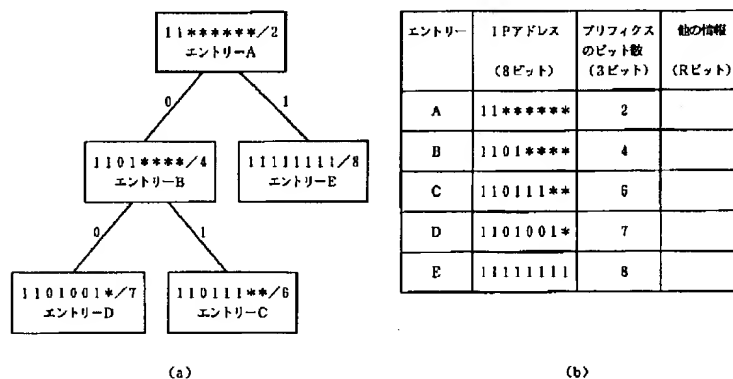
【符号の説明】

A、B、C、D、E、F IPアドレス

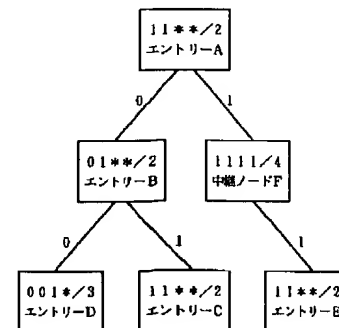
0E 検索ビットが0である子ノードの存在の有無を示す値

1E 検索ビットが1である子ノードの存在の有無を示す値

【図1】



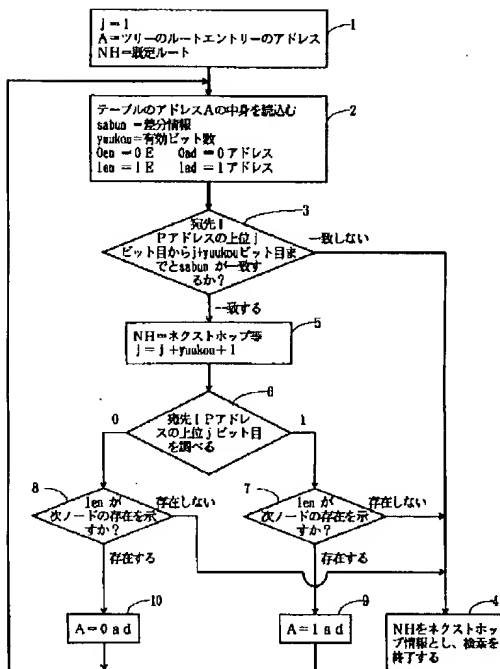
【図2】



【図3】

| エントリー | 差分情報<br>(4ビット) | 有効ビット数<br>(2ビット) | 0 E | 0 アドレス   | 1 E | 1 アドレス   | ネクスト<br>ホップ等 |
|-------|----------------|------------------|-----|----------|-----|----------|--------------|
| A     | 11**           | 2                | 1   | アドレス (B) | 1   | アドレス (F) |              |
| B     | 01**           | 2                | 1   | アドレス (D) | 1   | アドレス (C) |              |
| F     | 1111           | 4                | 0   |          | 1   | アドレス (E) |              |
| C     | 11**           | 2                | 0   |          | 0   |          |              |
| D     | 001*           | 3                | 0   |          | 0   |          |              |
| E     | 11**           | 2                | 0   |          | 0   |          |              |

【図4】



【図5】

